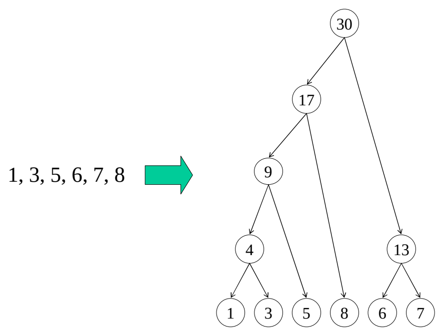
Data Structure Final (20070625)

---

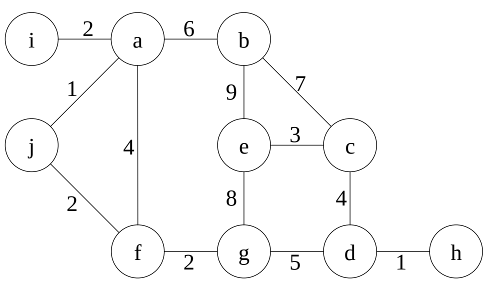
1. (15%)
   a. (3%) If a graph has 6 vertices, with the following degrees: 1, 1, 2, 2, 3, 5. How many edges does this graph have?
   Answer: (1+1+2+2+3+5)/2=7. (See page 262, section 6.1.2 of the text.)
   b. (3%) A graph G is represented by the following adjacency lists:
      - vertex 0 -> 1 -> 2
      - vertex 1 -> 0 -> 3 -> 4
      - vertex 2 -> 0 -> 5 -> 6
      - vertex 3 -> 1 -> 7
      - vertex 4 -> 1 -> 7
      - vertex 5 -> 2 -> 7
      - vertex 6 -> 2 -> 7
      - vertex 7 -> 3 -> 4 -> 5 -> 6

   What is the sequence of breadth first search starting from vertex 0?
   Answer: 0 1 2 3 4 5 6 7 (See page 274, section 6.2.2 of the text.)

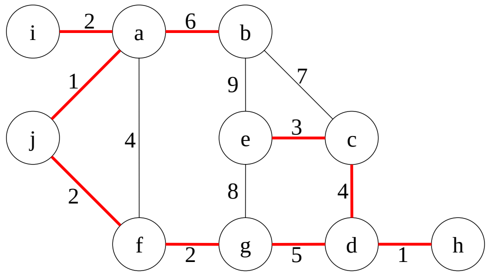
   c. (3%) What is the sequence of depth first search of the graph G in the previous sub-problem, starting from vertex 0?
   Answer: 0 1 3 7 4 5 2 6 (See page 274, section 6.2.2 of the text.)
   d. (3%) What is a "stable" sorting algorithm?
   Answer: A stable sorting algorithm will not chagne the relative order of already sorted elements. In symbol, if $a_i = a_j$ and $i < j$, then $\sigma_i < \sigma_j$ after a stable sorting algorithm.
   e. (3%) Given the weights of 1, 3, 4, 5, 7, 8, 12, construct a Huffman tree with minimum weighted external path length.
   Answer:



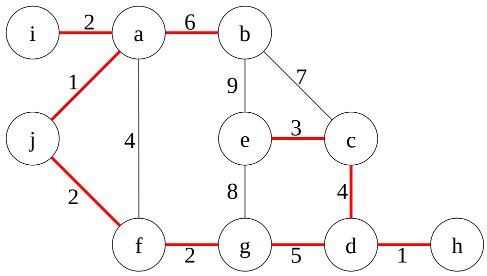
2. (6%) Answer the following short questions about insertion sort.
   a. (2%) What is the average and worse case time complexity of insertion sort of n numbers?
      Answer: $O(n^2)$, $O(n^2)$
   b. (2%) Given a random permutation of a sequence from 1 to 6, what is the sequence that exhibits the worst case behaviour of insertion sort?
      Answer: 6 5 4 3 2 1
   c. (2%) Is insertion sort stable?
      Answer: Yes.
3. (9%) Answer the following short questions about quick sort.
   a. (2%) What is the average and worse case time complexity of quick sort of n numbers?
      Answer: $O(n^2)$, O(n*log(n))
   b. (2%) Given a random permutation of a sequence from 1 to 6, what is the sequence that exhibits the worst case behaviour of quick sort?
      Answer: 1 2 3 4 5 6
   c. (3%) How do you avoid the worse case behavior of quick sort?
      Answer: Use the technique of "median of three" in page 332, section 7.4 of the text. Or you can do a random permutation before quick sort.
   d. (2%) Is quick sort stable?
      Answer: No.
4. (10%) Suppose that we are using 10 queues as the bins for LSD radix 10 sort. Show the contents of these queues after each of the three passes of LSD radix sort when the input sequence is [179, 208, 306, 93, 859, 984, 55, 9, 271, 33].
   Answer: See pages 356, 357, 358, Figure 7.16 of section 7.8 of the textbook.
5. (10%) Consider the following undirected graph:



a. (5%) Find a minimum-cost spanning tree by Kruskal's algorithm. (Please give partial results at the end of each iteration.)
Answer: (Click to see the answer.)



b. (5%) Repeat (a) using Prim's algorithm.
Answer: (Click to see the answer.)



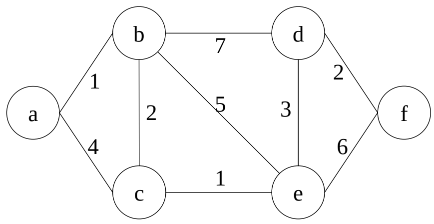
6. (10%)

a. (5%) Please describe Dijkstra's algorithm for the single-source all-destination shortest path problem. (Please clearly define your notation.)
Answer: (My description is a bit formal. You can use your own way to explain this algorithm.)
Let V denotes the set of all vertices. Let S denote the set of vertices, including the starting vertex a, whose shortest paths have been found. The distance between vertices x and y is denoted by w(x, y). The shortest distance from a to x is denoted by s(a, x).

1. Initially, set S={a}. For every vertex t in V-S, let s(t)=w(a, t).

2. Select a vertex t in V-S such that t = $\arg_y \min_{x \text{ in } S,\, y \text{ in } V\text{-}S}$ {s(x)+w(x, y)}
3. Include t in S. Repeat step 2 and 3 until the end vertex is included in S.

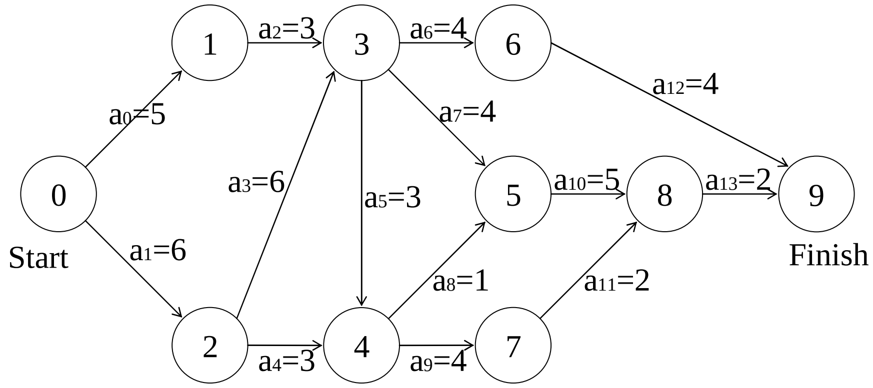
b. (5%) Please use the algorithm on the following graph to find the shortest paths from the source node a to all the other nodes. (Please show the partial result at each iteration.)



Answer: (You can also use other methods to demonstrate the operation of this algorithm.)

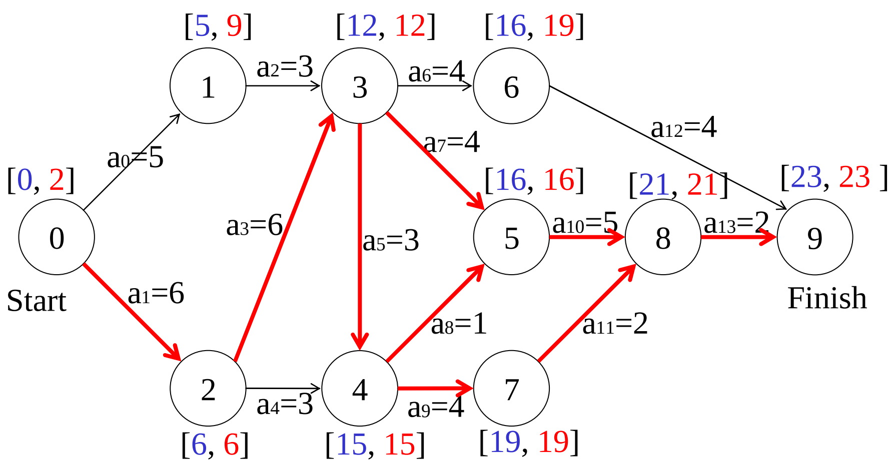
Iteration	S	Seleted vertex	Shortest distance a b c d e f
1	{a}	b	0 1 4 ∞ ∞ ∞
2	{a, b}	c	0 1 3 ∞ ∞ ∞
3	{a, b, c}	e	0 1 3 ∞ 4 ∞
4	{a, b, c, e}	d	0 1 3 7 4 ∞
5	{a, b, c, e d}	f	0 1 3 7 4 9

7. (10%) Consider the following AOE network:



a. (3%) Obtain the earliest and latest times for each vertex using the backward-forward approach.

Answer: (Earliest time in blue, latest time in red)



b. (3%) Obtain the early and late starting times for each activity.

Answer:

Activity	Early starting time	Late starting time	Critical activity
a_0	0	9-5=4	No
a_1	0	6-6=0	Yes
a_2	5	12-3=9	No
a_3	6	12-6=6	Yes
a_4	6	15-3=12	No
a_5	12	15-3=12	Yes
a_6	12	19-4=15	No
a_7	12	16-4=12	Yes
a_8	15	16-1=15	Yes
a_9	15	19-4=15	Yes
a_{10}	16	21-5=16	Yes
a_{11}	19	21-2=19	Yes
a_{12}	16	23-4=19	No
a_{13}	21	23-2=21	No

c. (2%) List all the critical paths.

Answer:

0. a_1, a_3, a_7, a_{10}, a_{13}
1. a_1, a_3, a_5, a_8, a_{10}, a_{13}
2. a_1, a_3, a_7, a_9, a_{11}, a_{13}

d. (2%) Can you reduce the project length by speeding up activity a_{10} alone? Why or why not?

Answer: No, you cannot reduce the project length by speeding up a_{10} alone. Besides speeding up a_{10}, you also need to speed up a_9 or a_{11} since they are parallel in two critical paths.

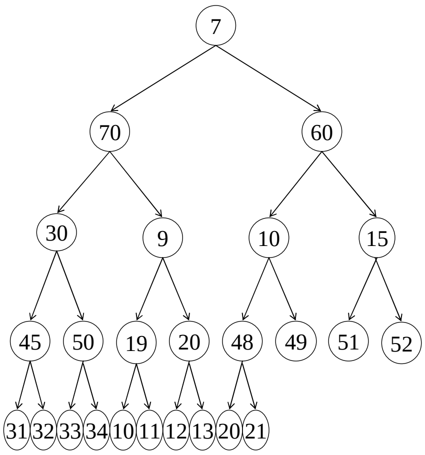
8. (10%)

a. (6%) Explain the operations of any 3 uniform hash functions that have been described in the textbook.

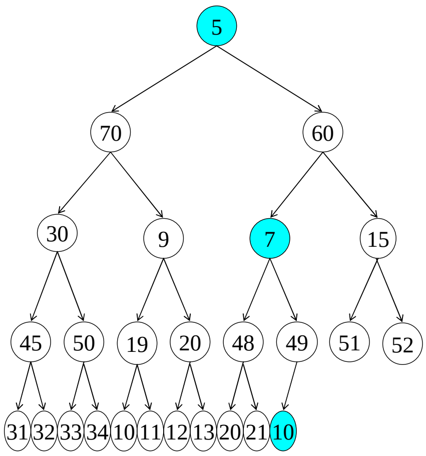
b. (1%) What is "collision" in accessing a hash table?

c. (3%) What is "overflow" in accessing a hash table? Describe 3 strategies in handling overflow?

9. (10%) Consider the following min-max heap:

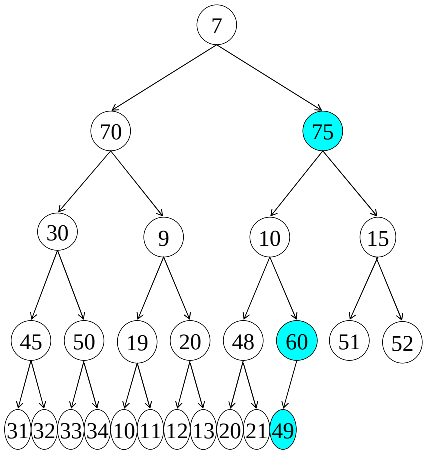


a. (2%) Plot the min-max heap after inserting 5 into the original heap.



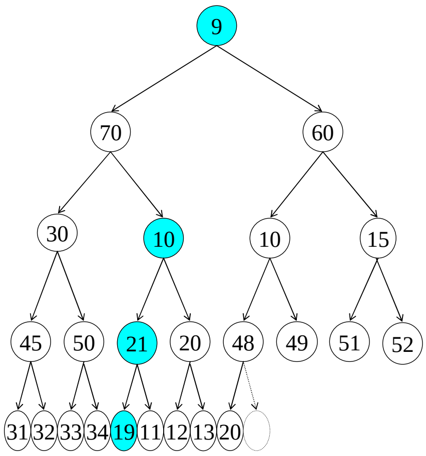
Answer:

b. (2%) Plot the min-max heap after inserting 75 into the original heap.



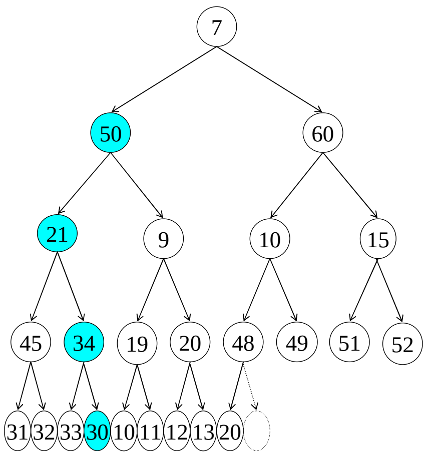
Answer:

c. (3%) Plot the min-max heap after deleting min from the original heap.



Answer:

d. (3%) Plot the min-max heap after deleting max from the original heap.

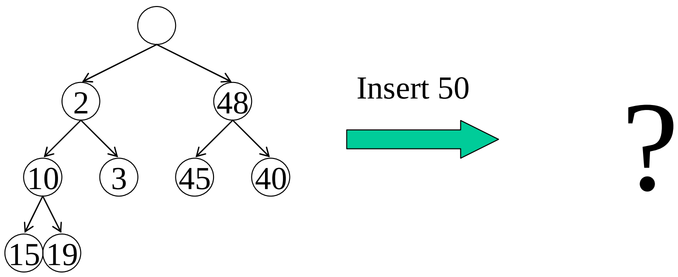


Answer:

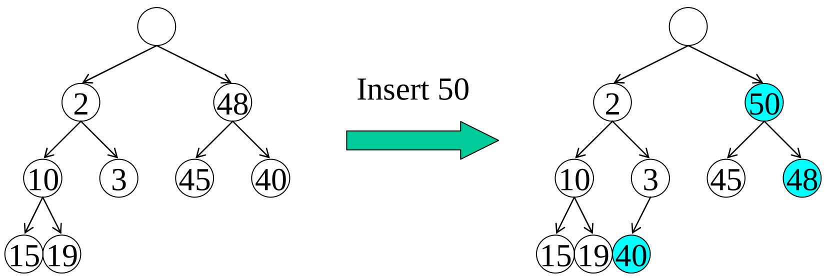
Please indicate the modified vertieces using double circles when drawing the final heaps.

10. (10%) Draw the deaps after the operations indicated next:

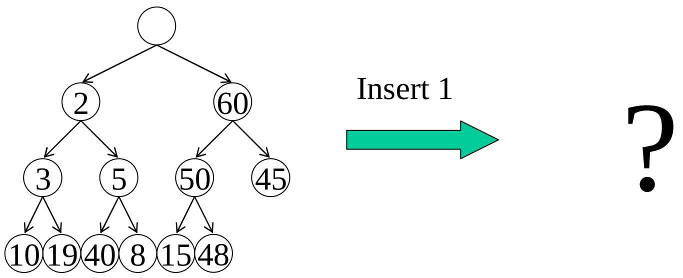
a. (2%)



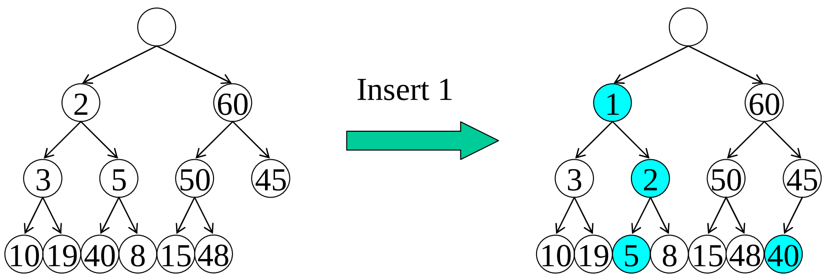
Answer:



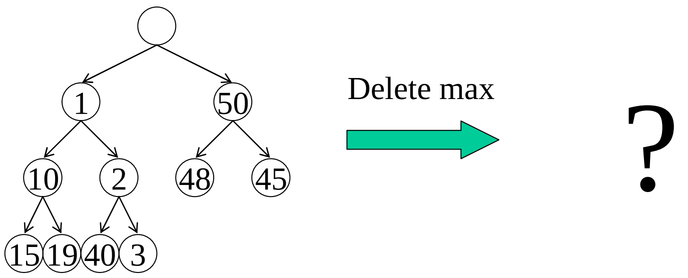
b. (2%)


2
60
3
5
50
45
10
19
40
8
15
48
Insert 1
?

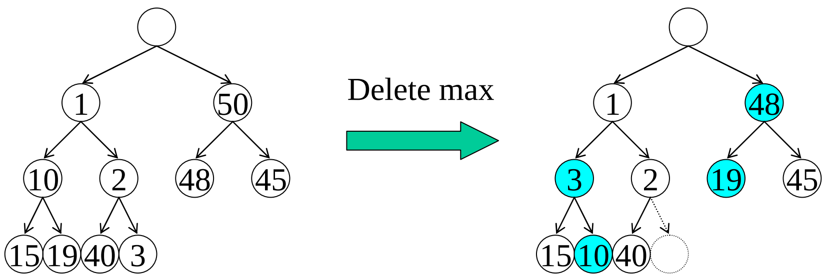
Answer:


2
60
3
5
50
45
10
19
40
8
15
48
Insert 1
1
60
3
2
50
45
10
19
5
8
15
48
40

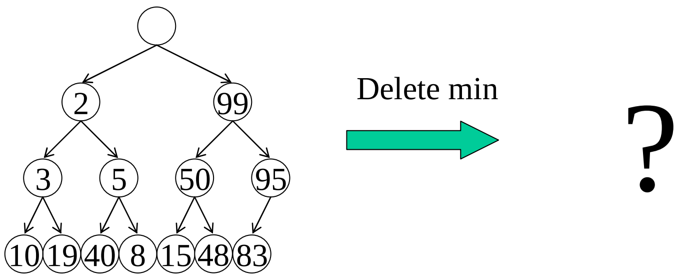
c. (3%)


1
50
10
2
48
45
15
19
40
3
Delete max
?

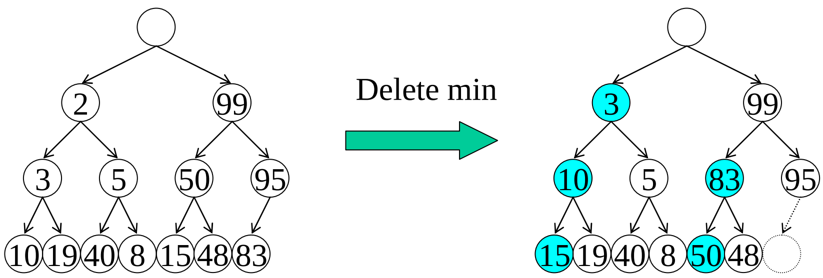
Answer:


1
50
10
2
48
45
15
19
40
3
Delete max
1
48
3
2
19
45
15
10
40

d. (3%)


2
99
3
5
50
95
10
19
40
8
15
48
83
Delete min
?

Answer:


2
99
3
5
50
95
10
19
40
8
15
48
83
Delete min
3
99
10
5
83
95
15
19
40
8
50
48